1. **Write a program for Abstraction?**

```java
abstract class Test {
    abstract String displayMessage(String name);

    abstract void displayMessage();
}

public class Abstraction extends Test {
    public static void main(String[] args) {
        Abstraction abstraction = new Abstraction();

        abstraction.displayMessage();
        System.out.println(abstraction.displayMessage("Test")); // Test

    }

    String displayMessage(String name) {
        return name;
    }

    void displayMessage() {
        System.out.println("Implementation for void method"); // Implementation for void method
    }
}
```

2. **Write an example for Encapsulation?**

```java
class Student {

    private int id;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

}

public class Encapsulation {

    public static void main(String[] args) {
        Student student = new Student();
        student.setId(10);

        System.out.println(student.getId()); //10
    }

}
```

3. **Write an example for Polymorphism?**

```
class C {

    public void display() {
        System.out.println("in Parent class Display");
    }
}

class D extends C {
    public void display() {
        System.out.println("in Parent class Display");
    }
}
public class Polymorphism {

    int sum(int i, int j) {
        return i + j;
    }

    int sum(int i, int j,int k) {
        return i + j + k;
    }

    public static void main(String[] args) {

        //Compile time Polymorphism
        Polymorphism polymorphism = new Polymorphism();
        System.out.println(polymorphism.sum(5, 10)); //15
        System.out.println(polymorphism.sum(4, 8,9)); //21

        //Runtime Polymorphisim
        C a = new C();
        a.display();
        D b = new D();
        b.display();
    }
}
```

4. **How to reverse a string without recursion?**

```
public static void main(String[] args) {
    String str = "Some Thing";

    String reverse = "";

    for (int i = str.length(); i > 0; --i) {
        reverse = reverse + (str.charAt(i-1));
    }

    System.out.println(reverse);//gnihT emoS

}
```

5. **How to swap two Strings without using a third variable?**

```
public static void main(String[] args) {
    String a = "abc";
    String b = "def";

    a = a + b;
    b = a.substring(0, a.length() - b.length());
    a = a.substring(b.length());

    System.out.println(a + " " + b); //def abc
}
```

6. **Write a program to reverse a number?**

```java
public static void main(String[] args) {
    int number = 126754;
    int reverse = 0;

    while (number != 0) {
        reverse = (reverse * 10) + (number % 10);
        number = number / 10;
    }

    System.out.println(reverse);//457621
}
```

7. **Write the example for Singleton class?**

```java
class SigleTonDemo {

    private static SigleTonDemo singleTon = new SigleTonDemo();

    private SigleTonDemo() {
    }

    public static SigleTonDemo getInstance() {
        return singleTon;
    }

    protected static void demoMethod() {
        System.out.println("demoMethod for singleton");
    }

}

public class SingleTon {

    public static void main(String[] args) {
        SigleTonDemo singleTon = SigleTonDemo.getInstance();
        singleTon.demoMethod(); //demoMethod for singleton

    }
```

8. **Write an example for Sting Sub String?**

```java
public static void main(String[] args) {

    String name = "TestingFramework";

    System.out.println(name.substring(3, 8)); //tingF

    name = "Testing Frame Work";

    System.out.println(name.substring(0, 9)); //Testing F
    System.out.println(name.substring(1, 9)); //esting F
}
```

9. **Write a program to check given input is palindrome or not?**

```java
public class PalindromExample {

    public static void main(String[] args) {

        String data = "aba";

        //Solution 1
        System.out.println(new StringBuilder().append(data).reverse().toString().equals(data));

        // Second Approach
        boolean isPalindrom = checkPalindrom(data);
        System.out.println(isPalindrom); // true
    }

    private static boolean checkPalindrom(String input) {
        String reverse = "";
        for (int i = input.length(); i > 0; --i) {
            reverse = reverse + (input.charAt(i - 1));
        }
        return reverse.equals(input);
    }

}
```

10. Write an example to know Even or Odd number?

```java
public static void main(String[] args) {
    int num =10;

    if(num%2 == 0) {
        System.out.println("Even"); //Even
    } else{
        System.out.println("Odd");
    }

    num =11;

    if(num%2 == 0) {
        System.out.println("Even");
    } else{
        System.out.println("Odd");//Odd
    }
}
```

11. Write an example to know given number is prime or not?

A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself. For example, 5 is prime, as only 1 and 5 divide it, whereas 6 is composite, since it has the divisors 2 and 3 in addition to 1 and 6.

```java
public static void main(String[] args) {

    System.out.println(isPrimeNumber(10)); //false
    System.out.println(isPrimeNumber(11)); //true
}

public static boolean isPrimeNumber(int number) {

    for (int i = 2; i <= number / 2; i++) {
        if (number % i == 0) {
            return false;
        }
    }
    return true;
}
```

12. Given an example to get the factorial of a number?

Example: if input is 3: 3*2*1 =6

```
public static void main(String[] args) {

    int fact = 1;
    int input = 3;

    for (int i = 1; i <= input; i++) {
        fact = fact * i;
    }

    System.out.println(fact); // 6
}
```

13. Write a logic to find the duplicate characters in String?

```
public static void main(String[] args) {

    String name = "SunnySy";

    Map<Character, Integer> dupCountMap = new HashMap<>();

    char[] nameArr = name.toCharArray();

    for (char c : nameArr) {
        if (dupCountMap.containsKey(c)) {
            dupCountMap.put(c, dupCountMap.get(c) + 1);
        } else {
            dupCountMap.put(c, 1);
        }
    }

    dupCountMap.forEach((k,v) -> System.out.println(k + " Count " + v)); //S Count 2 u Count 1  y Count 2 n Count 2
}
```

14. Write a program to check String is Anagram or not?

Two strings are called anagrams if they contain same set of characters but in different order.

```
public class AnagramExample {

    public static void main(String[] args) {
        boolean test = isAnagram("Keep", "Peek");
        System.out.println(test);
    }

    private static boolean isAnagram(String string, String string2) {
        char [] arr1 = string.toLowerCase().toCharArray();
        char [] arr2 = string2.toLowerCase().toCharArray();

        Arrays.sort(arr1);
        Arrays.sort(arr2);

        return Arrays.equals(arr1, arr2);
    }
```

15. Write a program to print fibonacci series?

By definition, the first two numbers in the Fibonacci sequence are 0 and 1, and each subsequent number is the sum of the previous two.

```
public static void main(String[] args) {
    int num1 = 0;
    int num2 = 1;
    int counter = 0;
    int input = 5;

    while (counter < input) {

        System.out.print(num1 + " "); //0 1 1 2 3

        int num3 = num2 + num1;
        num1 = num2;
        num2 = num3;
        counter = counter + 1;
    }
}
```

16. Write a program to print 1 to 100 and 100 to 1 numbers without loop?

```
// Sample Java Program to Print 1 to 100 without Loop
public class print1to100 {
        public static void main(String[] args)
        {
                int number = 1;

                printNumbers(number);
        }

        public static void printNumbers(int num)
        {
                if(num <= 100)
                {
                        System.out.print(num +" ");
                        printNumbers(num + 1);
                }
        }
}
```

```
// Sample Java Program to Print 100 to 1 without Loop
public class print1to100Ex2 {
        public static void main(String[] args)
        {
                printNumbers(100);
        }

        public static void printNumbers(int num)
        {
                if(num > 0)
                {
                        System.out.print(num +" ");
                        printNumbers(num - 1);
                }
        }
}
```

17. Write a logic to find the duplicate numbers and its count in an Array?

```
public static void main(String[] args) {

    var num= new int[] { 1, 2, 4, 9, 1, 2, 4, 3 };

    Map<Integer, Integer> dupMap = new HashMap<>();

    for (int i : num) {

        if (dupMap.containsKey(i)) {
            dupMap.put(i, dupMap.get(i) + 1);
        } else {
            dupMap.put(i, 1);
        }
    }

    dupMap.forEach((k,v) -> System.out.println(k +" Count "+ v)); //1 Count 2 2 Count 2 3 Count 1   4 Count 2 9 Count 1

}
```

18. Write a java program to rearrange Positive & Negative Values in an Array?

```
public class ArrayPositiveNegativeRearrange {

    public static void main(String[] args) {
        int[] arr = { 2, 4, -6, 8, -5, -10 };

        int j = 0;
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] < 0) { // if negative number found
                if (i != j) {
                    int temp = arr[i];
                    arr[i] = arr[j]; // swapping with leftmost positive
                    arr[j] = temp;
                }
                j++;
            }
        }

        Arrays.stream(arr).forEach(System.out::print); //-6 -5 -10 8 4 2
    }
}
```

19. Write a program to find Highest and Lowest numbers in an array?

```
public static void main(String[] args) {
    int numArr[] = new int[] { 1, 99, 34, 24, 65 };

    // Solution 1
    Arrays.sort(numArr);
    System.out.println("Highest Num " + numArr[numArr.length - 1] + " Lowest Num " + numArr[0]); // Highest Num 99

}
```

20. Give best example to compare and find common elements two arrays?

Below is the best way if we can manually compare it by writing the separate loops

```java
public static void main(String[] args) {

    int numArr[] = { 1, 99, 34, 24, 65 };
    int numArr2[] = { 1, 87, 34, 64, 65 };

    // Checking both arrays are Equal
    Object[] arr1 = { numArr };
    Object[] arr2 = { numArr2 };

    if (Arrays.deepEquals(arr1, arr2)) {
        System.out.println("Equal");
    } else {
        System.out.println("Not Equal"); // Not Equal
    }

    // Finding out two common elements in two arrays
    Set<Integer> setOne = new HashSet<>();
    Arrays.stream(numArr).forEach(item -> setOne.add(item));

    Set<Integer> setTwo = new HashSet<>();
    Arrays.stream(numArr2).forEach(item -> setTwo.add(item));

    setOne.retainAll(setTwo);

    System.out.println(setOne); //[1, 65, 34]
```

21. Write a program to remove duplicates from sorted array?

```java
public static void main(String[] args) {
    int[] arr = { 1, 2, 2, 3, 5, 4, 5, 4, 1 };

    Arrays.stream(arr).sorted().distinct().forEach(System.out::println); //1 2 3 4 5
}
```

22. Write a program to convert string to number without using Integer.parseInt() method?

```java
public static void main(String[] args) {
    String input = "123456";
    char[] chArr = input.toCharArray();
    int sum = 0;

    // get ascii value for zero
    int zeroAscii = (int) '0';
    System.out.println(zeroAscii); //48
    for (char c : chArr) {
        int tmpAscii = (int) c;
        sum = (sum * 10) + (tmpAscii - zeroAscii);
    }

    System.out.println(sum); // 123456
}
```

23. Write a program to create deadlock between two threads?

```
public static void main(String[] args) {
    String str1 = "Java";
    String str2 = "UNIX";

    Thread trd1 = new Thread("My Thread 1") {
        public void run() {
            while (true) {
                synchronized (str1) {
                    synchronized (str2) {
                        System.out.println(str1 + str2);
                    }
                }
            }
        }
    };

    Thread trd2 = new Thread("My Thread 2") {
        public void run() {
            while (true) {
                synchronized (str2) {
                    synchronized (str1) {
                        System.out.println(str2 + str1);
                    }
                }
            }
        }
    };

    trd1.start();
    trd2.start();
```

24. How to swap two numbers without using temporary variable?

```
public class MySwapingTwoNumbers {

    public static void main(String[] args) {
        int x = 10;
        int y = 20;

        x = x + y;
        y = x - y;
        x = x - y;

        System.out.println(x + " " + y); //20 10
    }

}
```

25. Write a program to implement hashcode and equals.

```java
class Price {
    public String name;
    public int id;

    Price(String name, int id) {
        this.name = name;
        this.id = id;
    }

    @Override
    public boolean equals(Object obj) {
        Price price = (Price) obj;
        return (price.name == this.name && price.id == this.id);

    }

    @Override
    public int hashCode() {
        return this.id;
    }
}

public class HashCodeAndEquals {

    public static void main(String[] args) {
        Price g1 = new Price("aa", 1);
        Price g2 = new Price("aa", 1);

        if (g1.hashCode() == g2.hashCode()) {
            if (g1.equals(g2))
                System.out.println("Both Objects are equal. "); //Both Objects are equal.
            else
                System.out.println("Both Objects are not equal. ");
        } else
            System.out.println("Both Objects are not equal. ");
    }

}
```

26. Give an example for Bubble Sort?

```java
public class BubbleSort {

    public static void main(String[] args) {
        int arr[] = { 3, 60, 35, 2, 45, 320, 5 };

        int n = arr.length;
        int temp = 0;

        for (int i = 0; i < n; i++) {
            for (int j = 1; j < n - i; j++) {
                if (arr[j - 1] > arr[j]) {
                    temp = arr[j - 1];
                    arr[j - 1] = arr[j];
                    arr[j] = temp;
                }
            }
        }

        Arrays.stream(arr).forEach(System.out::println); //2 3  5 35 45 60  320
```

27. Give an example for Binary Search?

```java
public class BinarySearch {
    public static void main(String[] args) {
        int arr[] = { 2, 3, 4, 10, 40 };
        int searchElement = 10;
        int left = 0;
        int right = arr.length - 1;

        int result = binarySearch(arr, left, right, searchElement);

        if (result == -1)
            System.out.println("Element not present");
        else
            System.out.println("Element found at index " + result);
    }

    static int binarySearch(int arr[], int left, int right, int x) {
        if (right >= left) {
            int mid = left + (right - left) / 2;
            System.out.println("mid " +mid);
            // If the element is present at the
            // middle itself
            if (arr[mid] == x)
                return mid;

            // If element is smaller than mid, then
            // it can only be present in left subarray
            if (arr[mid] > x)
                return binarySearch(arr, left, mid - 1, x);

            // Else the element can only be present
            // in right subarray
            return binarySearch(arr, mid + 1, right, x);
        }

        // We reach here when element is not present
        // in array
        return -1;
    }
```

28. Write a logic remove duplicates from ArrayList without using Set or Map?

```java
public class RemoveDuplicateList {

    public static void main(String[] args) {
        List<Integer> intList = new ArrayList<>();
        intList.add(1);
        intList.add(3);
        intList.add(3);
        intList.add(2);
        intList.add(1);

        intList.stream().distinct().forEach(System.out::println); //1 3 2
    }
```

29. Write a logic to remove the duplicate objects from List?

```java
public class RemoveDuplicateObjects {

    public static void main(String[] args) {
        Student student = new Student();
        student.setId(101);
        student.setName("Sunny");

        Student student1 = new Student();
        student1.setId(102);
        student1.setName("Bunny");

        Student student2 = new Student();
        student2.setId(101);
        student2.setName("Sunny");

        List<Student> studentList = new ArrayList<>();
        studentList.add(student);
        studentList.add(student1);
        studentList.add(student2);

        //Removing the duplicate objects
        Set<Student> distinctSet = studentList.stream()
                .collect(Collectors.toCollection(() -> new TreeSet<>(Comparator.comparing(Student::getId))));

        distinctSet.forEach(item -> System.out.println(item.getId() + " " + item.getName()));
    }
```

30. Given the list of employees, count number of employees with salary > 5000?

```java
public static void main(String[] args) {
    Student student = new Student();
    student.setId(101);
    student.setName("Sunny");
    student.setSalary(29000);;

    Student student1 = new Student();
    student1.setId(102);
    student1.setName("Bunny");
    student1.setSalary(5000);

    List<Student> studentList = new ArrayList<>();
    studentList.add(student);
    studentList.add(student1);

    //Removing the duplicate objects
    long salaryCount = studentList.stream().filter(emp -> emp.getSalary()>5000).count();
    System.out.println(salaryCount); //1
```

31. Given the list of employees, get the name of the employee with max and minimum salary?

```
Student student = new Student();
student.setId(101);
student.setName("Sunny");
student.setSalary(29000);

Student student1 = new Student();
student1.setId(102);
student1.setName("Bunny");
student1.setSalary(5000);

Student student2 = new Student();
student2.setId(103);
student2.setName("Munny");
student2.setSalary(1000);

List<Student> studentList = new ArrayList<>();
studentList.add(student);
studentList.add(student1);
studentList.add(student2);

//Max Salary
Student maxSalaryEmp = studentList.stream().max(Comparator.comparing(Student::getSalary)).get();
System.out.println(maxSalaryEmp.getName()); //Sunny

//Minimum Salary
Student minSalaryEmp =studentList.stream().min(Comparator.comparing(Student::getSalary)).get();
System.out.println(minSalaryEmp.getName()); //Munny
```

32. Write a logic to sort employee by his Name?

```
Student student = new Student();
student.setId(101);
student.setName("Sunny");
student.setSalary(29000);

Student student1 = new Student();
student1.setId(102);
student1.setName("Bunny");
student1.setSalary(5000);

Student student2 = new Student();
student2.setId(103);
student2.setName("Munny");
student2.setSalary(1000);

List<Student> studentList = new ArrayList<>();
studentList.add(student);
studentList.add(student1);
studentList.add(student2);

//Sorting by name
List<Student> sortedList = studentList.stream().sorted(Comparator.comparing(Student::getName)).collect(Collectors.toList());

sortedList.forEach(item -> System.out.println(item.getId() + "  " + item.getName()));//102  Bunny   103  Munny  101  Sunny
```

33. Given the list of employee, group them by employee name?

```
public static void main(String[] args) {
    Student student = new Student();
    student.setId(101);
    student.setName("Sunny");
    student.setSalary(29000);

    Student student1 = new Student();
    student1.setId(102);
    student1.setName("Bunny");
    student1.setSalary(5000);

    Student student2 = new Student();
    student2.setId(103);
    student2.setName("Munny");
    student2.setSalary(1000);

    List<Student> studentList = new ArrayList<>();
    studentList.add(student);
    studentList.add(student1);
    studentList.add(student2);

    //Sorting by name
    Map<String, List<Student>> sortedList = studentList.stream().collect(Collectors.groupingBy(Student::getName));

    sortedList.forEach((k,v) ->{
        v.forEach(data -> System.out.println("Name "+ k + " Empi Id " + data.getId() + " Emp Name " + data.getName()));
        /*
         Name Munny Empi Id 103 Emp Name Munny
         Name Bunny Empi Id 102 Emp Name Bunny
         Name Sunny Empi Id 101 Emp Name Sunny

         */
    });
```

34. Write a program to display the employee details in list descending order?

```
public static void main(String[] args) {
    List<Integer> intList = new ArrayList<>();
    intList.add(1);
    intList.add(9);
    intList.add(2);

    Collections.reverse(intList);

    System.out.println(intList);

    List<Integer> nwList = intList.stream().sorted().collect(Collectors.toList());

    System.out.println(nwList);

    List<Employee> empList = new ArrayList<>();
    Employee employee = new Employee();
    employee.setName("Bunny");
    employee.setSalary(1000);
    employee.setEmpId(100);
    empList.add(employee);

    Employee employee1 = new Employee();
    employee1.setName("Sunny");
    employee1.setSalary(2000);
    employee1.setEmpId(102);
    empList.add(employee1);

    Employee employee2 = new Employee();
    employee2.setName("Munny");
    employee2.setSalary(5000);
    employee2.setEmpId(101);
    empList.add(employee2);

    //Sorting Employee in descending order
    empList.sort(Comparator.comparing(Employee::getEmpId).reversed());

    empList.forEach(empData -> System.out.println(empData.getEmpId()));
```

35. Write a program to display the employee details in map descending order?

```
public static void main(String[] args) {
    Map<Integer, Employee> unsortMap = new HashMap<>();

    Employee employee = new Employee();
    employee.setName("Sunny");
    employee.setSalary(2000);
    employee.setEmpId(102);
    unsortMap.put(employee.getEmpId(), employee);

    Employee employee1 = new Employee();
    employee1.setName("Bunny");
    employee1.setSalary(1000);
    employee1.setEmpId(100);
    unsortMap.put(employee1.getEmpId(), employee1);

    Employee employee2 = new Employee();
    employee2.setName("Munny");
    employee2.setSalary(5000);
    employee2.setEmpId(101);
    unsortMap.put(employee2.getEmpId(), employee2);

    LinkedHashMap<Integer, Employee> sortedMapByKey = new LinkedHashMap<>();

    //Sorting Employee in descending order
    unsortMap.entrySet().stream().sorted(Map.Entry.comparingByValue(Comparator.comparing(Employee::getEmpId).reversed()))
            .forEachOrdered(data -> sortedMapByKey.put(data.getKey(), data.getValue()));

    sortedMapByKey.forEach((k,v)->System.out.println(v.getEmpId())); //102 101 100
```

36. Write a program to implement your own HashMap?

```
@Getter
@Setter
class MyEntry<K, V> {
    K key;
    V value;
    MyEntry<K, V> next;

    public MyEntry(K key, V value, MyEntry<K, V> next) {
        this.key = key;
        this.value = value;
        this.next = next;
    }

    @Override
    public int hashCode() {
        return (int) key;
    }

    @Override
    public boolean equals(Object obj) {
        return key.equals(obj);
    }
}

class HashMapImp<K, V> {
    List<MyEntry<K, V>> entryList = new ArrayList<>();

    public void put(K key, V value) {
        MyEntry<K, V> entry = new MyEntry<>(key, value, null);

        if (entryList.contains(entry.getKey())) {
            entryList.remove(entry.getKey());
        } else {
            entryList.add(entry);
        }
    }
```

```
public MyEntry<K, V> get(K key) {
    for (MyEntry<K, V> data : entryList) {
        if (data.getKey() == key) {
            return data;
        }
    }
    return null;
}
```

37. Write a logic to implement custom ArrayList?

```
public class CustomArrayList {
    private Object[] store;
    private int currentSize = 0;

    public CustomArrayList() {
        store = new Object[10];
    }

    public void add(Object input) {
        if (store.length - currentSize < 5) {
            store = Arrays.copyOf(store, store.length * 2);
        } else {
            store[currentSize++] = input;
        }
    }

    public Object get(int index) {
        Object data = null;
        if (index < currentSize) {
            data = store[index];
        } else {
            throw new ArrayIndexOutOfBoundsException();
        }
        return data;
    }

    public Object remove(int index) {
        if (index < currentSize) {
            store[index] = null;
        } else {
            throw new ArrayIndexOutOfBoundsException();
        }
        return store;
    }

    public int size() {
        return currentSize;
    }

public static void main(String[] args) {
    CustomArrayList mal = new CustomArrayList();
    mal.add(2);
    mal.add(5);
    for(int i=0;i<mal.size();i++){
        System.out.print(mal.get(i)+" "); //2 5
    }

    System.out.println("Element at Index 1:"+mal.get(1)); //Element at Index 1:5
    System.out.println("Removing element at index 1: "+mal.remove(1));

    for(int i=0;i<mal.size();i++){
        System.out.print(mal.get(i)+" "); //2 null
    }

}
```